Efficient decoupled pose estimation from a set of points

Omar Tahri, Helder Araujo, Youcef Mezouar, François Chaumette

▶ **To cite this version:**

Omar Tahri, Helder Araujo, Youcef Mezouar, François Chaumette. Efficient decoupled pose estimation from a set of points. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'2013, Nov 2013, Tokyo, Japan. pp.1608-1613. hal-00851997

HAL Id: hal-00851997

https://inria.hal.science/hal-00851997

Submitted on 19 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient decoupled pose estimation from a set of points

Omar Tahri[1] and Helder Araujo [2] and Youcef Mezouar [3] and François Chaumette [4]

***Abstract*— This paper deals with pose estimation using an iterative scheme. We show that using adequate visual information, pose estimation can be performed in a decoupling the estimation of translation and rotation. More precisely, we show that pose estimation can be achieved iteratively as a function of only three independent unknowns, which are the translation parameters. An invariant to rotational motion is used to estimate the camera position. Once the camera position is estimated, we show that the rotation can be estimated efficiently using a direct method. The proposed approach is compared against two classical methods from the literature. The results show that using our method, pose tracking in image sequences and the convergence rate for randomly generated poses are improved.**

I. Introduction

Pose estimation is a classical problem in computer vision [5], [15]. Nevertheless, there is a recent renewed interest as a result of automated navigation and model-based vision systems. For instance, pose can be used in pose-based visual servoing (PBVS) as input [23]. For image-based visual servoing (IBVS) as well, pose estimation can also be required to obtain the depth information for the computation of the interaction matrix involved in the control scheme. In practice, the behaviors of PBVS and IBVS are affected by the errors on the depth estimates, especially when the displacements to be performed are very large [17].

Pose estimation consists on the determination of the position and orientation of a camera with respect to an object coordinate frame using image information. Numerous methods to estimate pose have been proposed in the literature. They can be classified according to the features used or the nature of the estimation method. The geometric features considered for the estimation of the pose are often points [5], segments [6], contours, conics [18] or image moments [22]. Another important issue is the registration problem. Purely geometric [6], or numerical and iterative [5], [2], [16] approaches may be considered. Linear approaches give closed-form solutions free of initialization [7], [1], [14]. However, the estimated pose using such methods is sensitive to image noise and to errors on camera intrinsic parameters. Full-scale non-linear optimization techniques [16] minimize the error between the observation and the projection of the feature using the model, that is the reprojection error. The non-linear and iterative approaches have the advantage of being more accurate than the linear ones. On the other hand, their drawback is that they may be subject to local minima and, worse, divergence, if not correctly initialized. Furthermore, they usually require several iterations to minimize the cost function and generally they are more time consuming than the direct methods. These problems (i.e. local minima, divergence and time cost) are mainly due to non-linearities in the mapping between 3D and image space. The non-linearities are also usually the main reason for the failure of filtering strategies of the pose [13]. This occurs especially when the initial state is not accurate or when abrupt motions occur (for instance, for Extended Kalman Filter [21]).

In this paper, we deal with the selection of visual information that decreases the effect of the non-linearities between the variations in the image space and the 3D space. The contributions of this work are:

- We show that the iterative estimation of the pose can be expressed as the unconstrained minimization of a cost function on three unknowns only (the translation parameters).
- The visual features are chosen to minimize the non-linearities with respect to the camera position.
- Once the camera position is obtained using an iterative method, the rotation can be computed directly in the least-squares sense, that is, it is obtained without any iterative method. Therefore, the convergence speed and rate are only function of the translation.

The remaining of this paper is organized as follows: Section II recalls the pose problem and camera model used in the rest of the paper; Section III presents our pose estimation method and discusses its benefits; Section IV compares our method with two efficient iterative methods from the literature.

II. Pose estimation problem

Pose estimation consists in determining the rigid transformation ${}^c\mathbf{M}_o$ between the object frame $\mathcal{F}_o$ and the camera frame $\mathcal{F}_c$ in unknown position using the corresponding object image. It is well known that the relationship between an object point with coordinates $\mathbf{P}_c = [X_c,\, Y_c,\, Z_c,\, 1]^\top$ in $\mathcal{F}_c$ and $\mathbf{P}_o = [X_o,\, Y_o,\, Z_o,\, 1]^\top$ in $\mathcal{F}_o$ can be written:

$$\mathbf{P}_c = {}^c\mathbf{M}_o\,\mathbf{P}_o = \begin{bmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ \mathbf{0_{31}} & 1 \end{bmatrix} \mathbf{P}_o. \tag{1}$$

The matrix ${}^c\mathbf{M}_o$ can be estimated by minimizing the modulus of the error in the image:

$$e = \| \mathbf{s}({}^c\mathbf{M}_o) - \mathbf{s}^* \|, \tag{2}$$

[1,2] O. Tahri and H. Araujo are with the Institute of Systems and Robotics of Coimbra, Portugal omartahri@isr.uc.pt, helder@isr.uc.pt

[3] Y. Mezouar is with Clermont Université, IFMA, Institut Pascal, BP 10448, F-63000 Clermont-Ferrand,France youcef.mezouar@ifma.fr

[4] F. Chaumette is with Inria Rennes-Bretagne Atlantique, Campus de Beaulieu, 35 042 Rennes-cedex, France (Francois.Chaumette@irisa.fr).

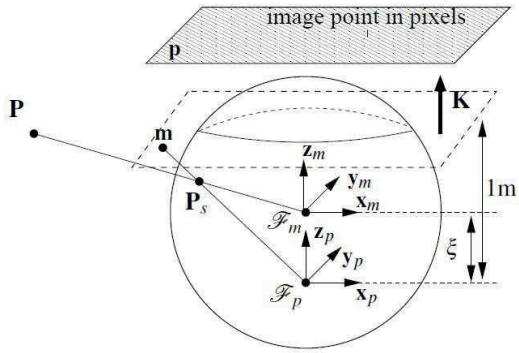

Fig. 1. Unified image formation

where $\mathbf{s}^*$ is the value of a set of visual features computed in the image acquired with the camera in unknown position and $\mathbf{s}({}^{\mathbf{c}}\mathbf{M}_{\mathbf{o}})$ is the value of the same set of features computed from the object model, the transformation ${}^c\mathbf{M}_o$, and the camera model. In the following paper, we consider the case of central cameras. A unified model for central imaging systems has been proposed in [8]. It consists in modeling the central imaging systems by two consecutive projections: spherical and then perspective. Consider $\mathcal{F}_m$ the frame attached to a virtual unitary sphere as shown on Fig. 1. The frames attached to the sphere $\mathcal{F}_m$ and to the perspective camera $\mathcal{F}_p$ are related by a simple translation of $-\xi$ along the Z-axis. Let $\mathbf{P}$ be a 3D point with coordinates $\mathbf{P} = (X, Y, Z)$ in $\mathcal{F}_m$. The world point $\mathbf{P}$ is projected onto:

$$\mathbf{m} = \begin{pmatrix} x, & y, & 1 \end{pmatrix} = \begin{pmatrix} \frac{X}{Z+\xi\|\mathbf{P}\|}, & \frac{Y}{Z+\xi\|\mathbf{P}\|}, & 1 \end{pmatrix} \quad (3)$$

and the coordinates of the projected points in the image plane are obtained after a plane-to-plane collineation $\mathbf{K}$: $\mathbf{p} = \mathbf{K}\mathbf{m}$, ($\mathbf{K}$ is a 3×3 matrix containing the camera intrinsic parameters). The matrix $\mathbf{K}$ and parameter ξ can be obtained from calibration using, for example, the methods proposed in [8]. In the sequel, the imaging system is assumed to be calibrated. In this case, the inverse projection onto the unit sphere can be obtained from:

$$\mathbf{P_s} = \gamma \begin{pmatrix} x, & y, & 1 - \frac{\xi}{\gamma} \end{pmatrix} \quad (4)$$

where

$$\gamma = \frac{\xi + \sqrt{1 + (1-\xi^2)(x^2+y^2)}}{1+x^2+y^2}.$$

The projection onto the unit sphere from the image plane is possible for all sensors obeying the unified model. In other words, it encompasses all sensors in this class namely [9]: perspective and catadioptric cameras. A large class of fisheye cameras can also be represented by this model [4], [3].

III. Pose estimation method

In this section, we first propose new features to estimate the camera position separately from the rotation. We then present a method for the direct estimation of the rotation once the translational part of the pose has been determined.

A. Position estimation using an invariant to rotation

1) Invariant to rotations: Let d_{ij} be the distance between two projected points $\mathbf{P}_{s_i}$ and $\mathbf{P}_{s_j}$ on the unit sphere

$$d_{ij} = \sqrt{2 - 2\mathbf{P}_{s_i}^\top \mathbf{P}_{s_j}} \quad (5)$$

It can be easily shown that the distance d_{ij} is an invariant to any rotational motion applied to the camera frame. Therefore, the variation of d_{ij} only depends of the translation. Furthermore, the Jacobian matrix that links the variation of d_{ij} with respect to translational displacement is given by:

$$\mathbf{J}_{d_{ij}} = -\frac{\mathbf{P}_{s_i}^\top \mathbf{J}_{\mathbf{P}_{s_j}} + \mathbf{P}_{s_j}^\top \mathbf{J}_{\mathbf{P}_{s_i}}}{d_{ij}} \quad (6)$$

where $\mathbf{J}_{\mathbf{P}_{s_i}}$ and $\mathbf{J}_{\mathbf{P}_{s_j}}$ are the Jacobian matrices that relate the variation of the point coordinates on the unit sphere to the camera translational displacements. This Jacobian can be written as [11]:

$$\mathbf{J}_{\mathbf{P}_{s_i}} = \frac{-\mathbf{I} + \mathbf{P}_{s_i}\mathbf{P}_{s_i}^\top}{\| \mathbf{P_i} \|} \quad (7)$$

where $\| \mathbf{P_i} \|$ is the distance of the 3D point to the center of the sphere. After inserting (7) in (6), we obtain:

$$\mathbf{J}_{d_{ij}} = -\frac{1}{d_{ij}} \left((-\frac{1}{\| \mathbf{P_j} \|} + \frac{\mathbf{P}_{s_i}^\top \mathbf{P}_{s_j}}{\| \mathbf{P_i} \|}) \mathbf{P}_{s_i}^\top + (-\frac{1}{\| \mathbf{P_i} \|} + \frac{\mathbf{P}_{s_i}^\top \mathbf{P}_{s_j}}{\| \mathbf{P_j} \|}) \mathbf{P}_{s_j}^\top \right) \quad (8)$$

Further to the invariance to rotation, which allows separating the estimation of the camera position and orientation, it is also possible to decrease the non-linearities between the image space and 3D space. Indeed, the distance d_{ij} behaves as a function which is approximately inversely proportional to the point depths $\| \mathbf{P_i} \|$. As it can be seen in (8), its corresponding Jacobian matrix depends on the square of the inverse of the point depths. On the other hand, the inverse of the distance behaves approximately as a linear function of the points depths. This allows obtaining nice linearizing properties between the image space and 3D space. We propose thus to use $s_{ij} = 1/d_{ij}$ for all possible combinations of two projected points. In the next section, we show how to take into account the noise propagation from the image space to the new feature space.

2) Noise propagation from image space to the new feature space: Let us first see how noise in the image plane acts on a projected point onto the sphere. Taking the derivative of (4) and using a first order approximation, the variation of the coordinates of the point projected onto the sphere can be written as a function of the variation of the coordinates of the image points:

$$\Delta\mathbf{P}_s = \mathbf{J}_{\mathbf{P}_s/\mathbf{m}} \Delta\mathbf{m} \quad (9)$$

where:

$$\mathbf{J}_{\mathbf{P}_s/\mathbf{m}} = \begin{bmatrix} \gamma + x\frac{\partial\gamma}{\partial x} & x\frac{\partial\gamma}{\partial y} & 0 \\ y\frac{\partial\gamma}{\partial x} & \gamma + y\frac{\partial\gamma}{\partial y} & 0 \\ \frac{\partial\gamma}{\partial x} & \frac{\partial\gamma}{\partial y} & 0 \end{bmatrix} \quad (10)$$

with:

$$\frac{\partial \gamma}{\partial x} = \frac{x}{1+x^2+y^2}\left(\frac{(1-\xi^2)}{(\sqrt{1+(1-\xi^2)(x^2+y^2)}} - 2\gamma\right)$$
$$\frac{\partial \gamma}{\partial y} = \frac{y}{1+x^2+y^2}\left(\frac{(1-\xi^2)}{(\sqrt{1+(1-\xi^2)(x^2+y^2)}} - 2\gamma\right) \quad (11)$$

where γ and ξ have been defined in Section II. Therefore, the variation of $\mathbf{P}_s$ with respect to image points in pixels is given by:

$$\Delta \mathbf{P}_s = \mathbf{J}_{\mathbf{P}_s/\mathbf{m}} \mathbf{K}^{-1} \Delta \mathbf{p} \quad (12)$$

Furthermore, from $d_{ij} = \sqrt{2 - 2\mathbf{P}_{si}^\top \mathbf{P}_{sj}}$, we have:

$$\Delta d_{ij} = -\frac{1}{d_{ij}} (\mathbf{P}_{sj}^\top \Delta \mathbf{P}_{si} + \mathbf{P}_{si}^\top \Delta \mathbf{P}_{sj}) \quad (13)$$

As a result of (10) and (13), the variation of $s_{ij} = \frac{1}{d_{ij}}$ with respect to noise in the coordinates of the image points (in pixels) is obtained by:

$$\Delta s_{ij} = \mathbf{J}_{s_{ij}/\mathbf{p}'} \begin{bmatrix} \Delta \mathbf{p}_i \\ \Delta \mathbf{p}_j \end{bmatrix} \quad (14)$$

where $\mathbf{J}_{s_{ij}/\mathbf{p}} = \begin{bmatrix} \mathbf{P}_{sj}^\top \mathbf{J}_{\mathbf{P}_{si}/\mathbf{m_i}} \mathbf{K}^{-1} & \mathbf{P}_{si}^\top \mathbf{J}_{\mathbf{P}_{sj}/\mathbf{m_j}} \mathbf{K}^{-1} \end{bmatrix} / d_{ij}^3$. In order to take into account the effect of the mapping from the image point coordinates to the features s_{ij}, each visual feature should be weighted by $\frac{1}{\|\mathbf{J}_{s_{ij}^*/\mathbf{p}^*}\|}$ computed using the image points coordinates corresponding to the pose to be computed. More precisely, we use all possible combinations of $s_{wij} = \frac{1}{d_{ij}} \frac{1}{\|\mathbf{J}_{s_{ij}^*/\mathbf{p}^*}\|}$ as measure to estimate the camera position. The iterative algorithm to estimate the translation will be described in III-C. In the next paragraph, we use a direct method to estimate the orientation of the camera.

B. Direct estimation of the rotation by solving an orthogonal Procrustes problem

After estimating the translation using $s_{wij} = \frac{1}{d_{ij}} \frac{1}{\|\mathbf{J}_{s_{ij}^*/\mathbf{p}^*}\|}$ and removing it from the pose, the rotation matrix can be directly estimated in one step by solving an orthogonal Procrustes problem between the two sets of projected points on the sphere. We recall that the Orthogonal Procrustes problem is defined as the least squares problem transforming a given matrix $\mathbf{F}$ into a given matrix $\mathbf{F}'$ by an orthogonal transformation $\mathbf{R}$ so that the sum of squares of the residual matrix $\mathbf{E} = \mathbf{RF} - \mathbf{F}'$ is minimal [12]. In our context, the matrices $\mathbf{F}$ and $\mathbf{F}'$ are composed by the set of all projected points onto the unit sphere:

$$\mathbf{F}' = [\mathbf{P}_{s1}^* \ \mathbf{P}_{s2}^* \dots \mathbf{P}_{sN}^*],$$

and

$$\mathbf{F} = [\mathbf{P}_{s1} \ \mathbf{P}_{s2} \dots \mathbf{P}_{sN}],$$

The Orthogonal Procrustes Problem can be solved by computing the SVD decomposition of $\mathbf{F}'\mathbf{F}^\top$ [19]:

$$\mathbf{F}'\mathbf{F}^\top = \mathbf{U}\Sigma\mathbf{V}^\top \quad (15)$$

The rotation matrix between the two camera poses is then given by:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top \quad (16)$$

C. Pose estimation algorithm

As already said, the pose estimation method is divided into two steps: firstly, we determine the translation between an initial pose and the pose to be estimated using the invariant to rotation as feature as follows:

- Project the image points corresponding to the pose to be computed onto the sphere using (4).
- Compute the value of features vector $\mathbf{s_t}^*$ for the pose to be estimated by stacking the features $s_{wij}^* = \frac{1}{d_{ij}^*} \frac{1}{\|\mathbf{J}_{s_{ij}^*/\mathbf{p}^*}\|}$.
- The camera pose is set up at an initial value:

$${}^c\mathbf{M}_o = {}^i\mathbf{M}_o = \begin{bmatrix} {}^i\mathbf{R}_o & {}^i\mathbf{t}_o \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$$

Minimization loop: while ($\| \mathbf{s_t} - \mathbf{s_t}^* \| \leq \varepsilon$) where ε is defined by the user.

- Project the 3D points of the object onto the unit sphere using the object model and the current value of the pose ${}^c\mathbf{M}_o$.
- Compute the current value of features vector $\mathbf{s_t}$ corresponding to $\mathbf{s_t}^*$ by stacking the features $s_{wij} = \frac{1}{d_{ij}} \frac{1}{\|\mathbf{J}_{s_{ij}^*/\mathbf{p}^*}\|}$.
- Compute the Jacobian matrix $\mathbf{J}_{\mathbf{s_t}}$ corresponding to $\mathbf{s_t}$ ($\mathbf{J}_{\mathbf{s_t}}$ is an $l \times 3$ matrix, l is the number of used distances between projected points on the sphere).
- Compute the translational displacement using $\Delta\mathbf{t} = -\lambda \mathbf{J}_{\mathbf{s_t}}^+ (\mathbf{s_t} - \mathbf{s_t}^*)$ (λ is a scalar gain that tunes the convergence speed and $\mathbf{J}_{\mathbf{s_t}}^+$ is the pseudo-inverse of $\mathbf{J}_{\mathbf{s_t}}$)
- Update ${}^c\mathbf{M}_o$ by adding the translational motion $\Delta\mathbf{t}$.

Once the minimization loop described above has been achieved, the matrix ${}^c\mathbf{R}_i$ that defines the rotation between the initial camera pose (defined by ${}^i\mathbf{M}_o$) and the camera pose to be computed can be directly obtained from the direct method presented in Section III.B. This means that if the translational motion is well estimated using an invariant to rotations, the correct pose will be obtained. Note that the iterative minimization process is an optimization procedure without constraints of a cost function on three unknowns only, which is a significant advantage. Therefore, the convergence speed and rate are only function of the translations. This pose estimation algorithm can be considered a mixed method since translation is estimated iteratively whereas rotation is estimated in one step, in the least-squares sense.

IV. VALIDATION RESULTS

In this part, our pose estimation method is compared to two non-linear and iterative methods proposed respectively in [2] (method A in the following) and in [16] (method L in the following). The method L is a globally convergent algorithm that minimizes error in object space: the error between the observation and the projection of the features using the model. On the other hand, the method A minimizes an error defined in the image and improves the classical Lowe's pose-estimation algorithm. A comparison of several iterative methods has been made in [10] and showed that the method A is the most accurate of the considered methods.

A. Results for pose tracking

In this paragraph, the ability of each method to track the pose of the camera with respect to a set of points for image sequences with abrupt motions is tested. A camera model with focal scaling factors $F_x = F_y = 800\ pixels/m$ and principal point coordinates $u_x = v_x = 400\ pixels$ has been used to compute the image points. For our method, the scalar gain λ has been set to 1.

The first sequence of 300 images is obtained using 9 non coplanar points defined in the object frame by:

$$\mathbf{X}_1 = \begin{bmatrix} 0.2 & -0.2 & -0.2 & 0.2 & 0 & 0 & 0.1 & -0.13 & 0.4 \\ 0.2 & -0.2 & 0.2 & -0.2 & 0 & 0.15 & 0.01 & 0 & 0.4 \\ 1.01 & 1.02 & 0.96 & 1.03 & 1. & 1. & 1. & 1.2 & 1.3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{17}$$

White Gaussian noise with standard deviation equal to 0.5 has been added to the coordinates of each point in the image. Furthermore, the identity matrix has been used to initialize ${}^i\mathbf{M}_o$ for the first image of the sequence (the initial set of points is assumed to be in front of the camera close to the optical axis and at 1 *meter* distance from the image plane). The computed pose for each image is used as initialization to determine the pose for the following one using each method. The evolution of the real pose parameters of the camera with respect to the object frame is shown in Fig. 2. Let us consider the pose error defined by:

$$\mathbf{T_e} = \begin{bmatrix} \mathbf{R_e} & \mathbf{t_e} \\ \mathbf{0_{1\times 3}} & 1 \end{bmatrix} = \mathbf{T_r^{-1}T_c}, \tag{18}$$

where $\mathbf{T_r}$ and $\mathbf{T_c}$ are respectively the real and the estimated poses. If the correct pose is obtained, $\mathbf{T_e}$ is equal to the identity matrix ($\| \mathbf{t_e} \| = 0$ and $\mathbf{R_e} = \mathbf{I_3}$). Let θ_e be the norm of the rotation vector $\theta_e\mathbf{u}$ corresponding to the rotation matrix $\mathbf{R_e}$ (recall that $\theta_e\mathbf{u}$ is linked to $\mathbf{R_e}$ by the Rodrigues' formula). The errors $\| \mathbf{t_e} \|$ and θ_e on the estimated poses using our method, method A and method L are shown respectively in Figs 3, 4 and 5. From these plots, it can be seen that the estimated values using the three methods are similar and close to the real ones. Furthermore, the errors on the estimated pose obtained using the three methods are similar.

The second image sequence is obtained using less points (5 non-coplanar points):

$$\mathbf{X}_2 = \begin{bmatrix} 0.2 & -0.2 & -0.2 & 0.2 & 0 \\ 0.2 & -0.2 & 0.2 & -0.2 & 0.4 \\ 1.01 & 1.01 & 0.95 & 1.03 & 1. \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{19}$$

A stronger white gaussian noise with standard deviation equal to 2 has been added to the coordinates of each point. The results obtained using our method, method A and method L are shown respectively in Figs 7 and 8. The graphs obtained using method A are not shown since the algorithm diverged. From Fig. 7, it can be seen that the estimated values of the pose using our method follow closely the real ones. Finally, as it was mentioned in [20], method L is affected by local minima. Indeed from the plots, it can be noticed that the pose switched several times to local minima (refer to Fig. 8).

B. Convergence for random poses

In this paragraph, we compare the convergence rate for random poses using our method, method *L* and method *A*. The following setup has been used:

- An object composed of 8 coplanar points defined as follows has been considered:

$$\mathbf{X}_3 = \begin{bmatrix} -0.4 & 0.4 & -0.4 & 0.4 & 0.42 & -0.09 & 0.32 & -0.32 \\ -0.4 & -0.4 & 0.4 & 0.4 & -0.28 & 0.32 & 0 & 0 \\ 1. & 1 & 1. & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{20}$$

- Random poses have been generated as follows:
  - 1000 random rotational motions are firstly applied to the point coordinates defined in the object frame. The norm of the rotation around the x-axis and the y-axis range from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, while the rotation angle around the optical axis ranges from 0 to 2π.
  - for each generated rotation, a translational motion with respect to the optical axis that ranges from 1 *meter* to 4 *meters* is applied to the point coordinates defined in the object frame. Furthermore, the translational motions with respect to the x-axis and the y-axis are chosen such that the points coordinates belong to the image boundaries $[1\ 800; 1\ 800]\ pixels$.

The errors on pose are calculated using $\| \mathbf{t_e} \|$ and θ_e computed from (18). Furthermore, for all methods, the identity matrix is used as the initial value of the pose matrix. Figures 9.a and 9.b give the distributions of $\| \mathbf{t_e} \|$ and θ_e using the three different methods and using perfect data (no noise on the point coordinates in the image). In other words, for each value of $\| \mathbf{t_e} \|$ and θ_e, the plot gives the percentage of the errors smaller or equal to these values. From these figures, it can be seen that our method achieves a convergence rate around 90%, while method *L* and *A* achieve convergence rates around 70% and 50% respectively. The case of non convergence to the global minimum using our method and method *L* are due to convergence to local minima. Conversely, in the case where the method *A* is used, the non convergences to the global minimum are due to both divergence and convergence to local minima.

Next, we test the convergence rate of the three methods using the same setup, but with 1 pixel standard deviation gaussian noise on the point coordinates in the image. The results obtained using each method are given on Fig. 10. From this figure, it can be noticed that the accuracy of all the pose estimation methods decreased. However, our iterative method gives more accurate estimates for the poses.

As it has been shown above, pose estimation can be performed as an iterative minimization without constraints for only three parameters that are the translation parameters t_x, t_y and t_z. This limits the space that has to be searched for to find the global optimum. Since the method we propose allows

a high convergence rate, this makes it possible preinitializing the iterative algorithm at random starting points. The low dimensionality of the space permits also to visualize the local minima. Let us consider the following case where the object point coordinates are defined by (21) and the translational motion to be computed is defined by the vector $[0.7\ 0.4\ 2]m$. Fig. 11 shows the cost function $\| \mathbf{s}_t^* - \mathbf{s}_t \|$ as color level for 5 values of t_z and t_x and t_y ranging from $-2m$ to $+2m$. From this figure, we can visualize the positions of the local and global minima with respect to each other (the position of the minimal value of the cost function for each value t_z is marked by a cross in the images). From Fig 11.c ($t_z = 2m$), it can seen that the global minimum corresponds well to ($t_x = 0.7m$) and ($t_y = 0.4m$). We also note from Figs 11.d and 11.e that no local minima exist for $t_z > 2m$.

$$\mathbf{X}_4 = \begin{bmatrix} -0.4 & 0.4 & -0.4 & 0.4 & 0.5 \\ -0.4 & -0.4 & 0.4 & 0.4 & 0.6 \\ 1. & 1 & 1. & 1 & 1 \end{bmatrix} \qquad (21)$$

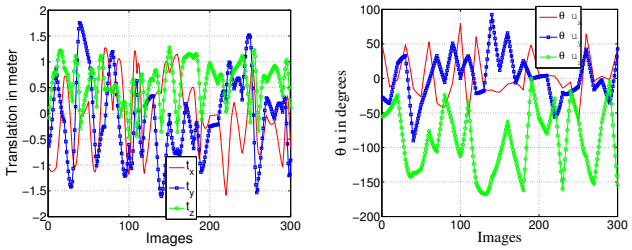

Fig. 2. Real values of the pose for the image sequence 1 versus image number: left) translation vector entries in meter, right) rotation vector entries in degrees.

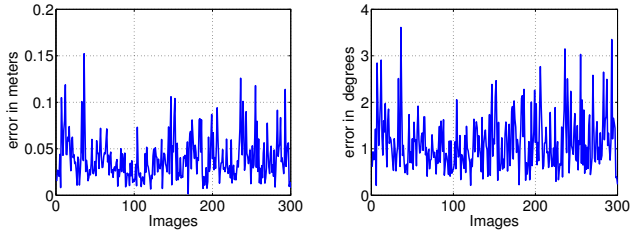

Fig. 3. Error on the estimated pose parameters using our method for the image sequence 1 versus image number: left) $\| \mathbf{t_e} \|$, right) θ_e.

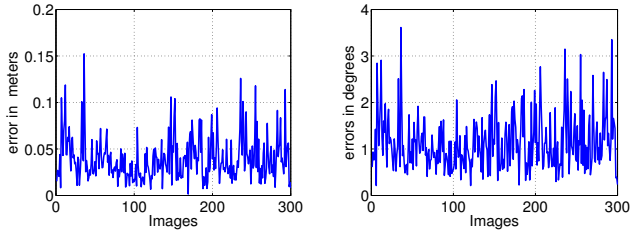

Fig. 4. Error on the estimated pose parameters using method A for the image sequence 1 versus image number: left) $\| \mathbf{t_e} \|$, right) θ_e.

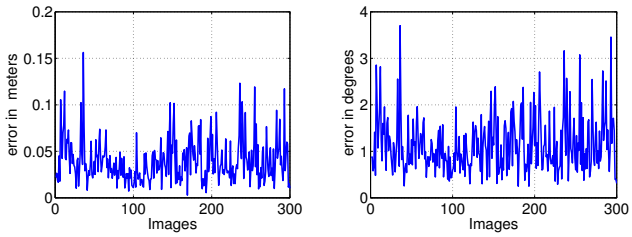

Fig. 5. Error on the estimated pose parameters using method L for the image sequence 1 versus image number: left) $\| \mathbf{t_e} \|$, right) θ_e.

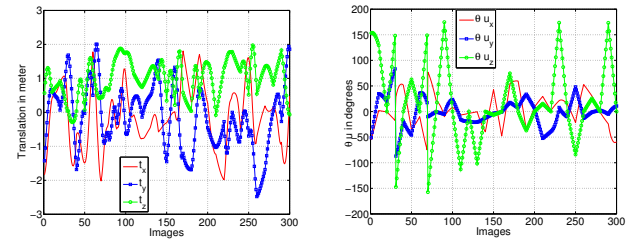

Fig. 6. Real values of the pose for the image sequence 2 versus image number: left) translation vector entries in meter, right) rotation vector entries in degrees.

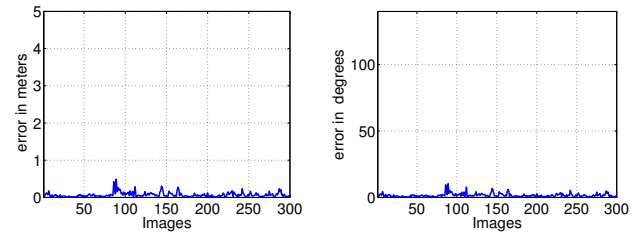

Fig. 7. Error on the estimated pose parameters using our method for the image sequence 2 versus image number: left) $\| \mathbf{t_e} \|$, right) θ_e.

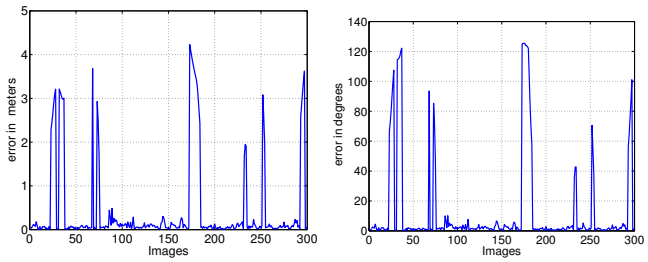

Fig. 8. Error on the estimated pose parameters using method L for the image sequence 1 versus image number: left) $\| \mathbf{t_e} \|$, right) θ_e.

V. Conclusion and Future Work

In this paper, we have proposed a new pose estimation method from a set of matched points based on an invariant to rotations. The method has a mixed nature in the sense that only translation is estimated iteratively, which is possible as a result of using an invariant to rotation. Its mixed nature, and the fact the iterative optimization is used to estimated only three unknowns, allows for its robustness and accuracy. The proposed method has been validated and compared to two different non-linear methods. The results obtained show

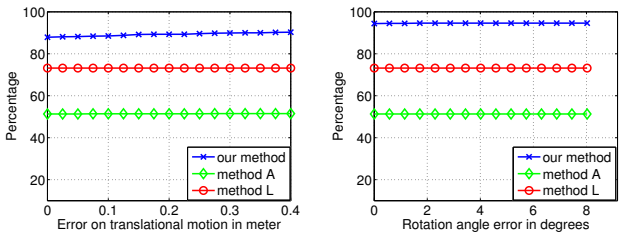

Fig. 9. Percentage of convergence with perfect data: a) $\| \mathbf{t_e} \|$, b) θ_e

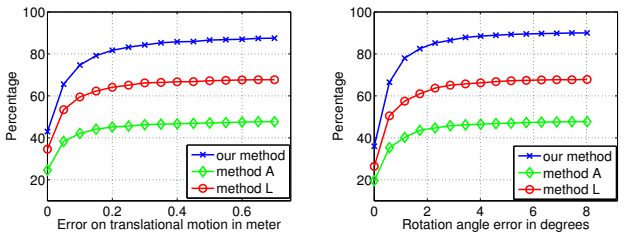

Fig. 10. Percentage of convergence with 1 pixel gaussian noise on image point coordinates: left) $\| \mathbf{t_e} \|$, right) θ_e

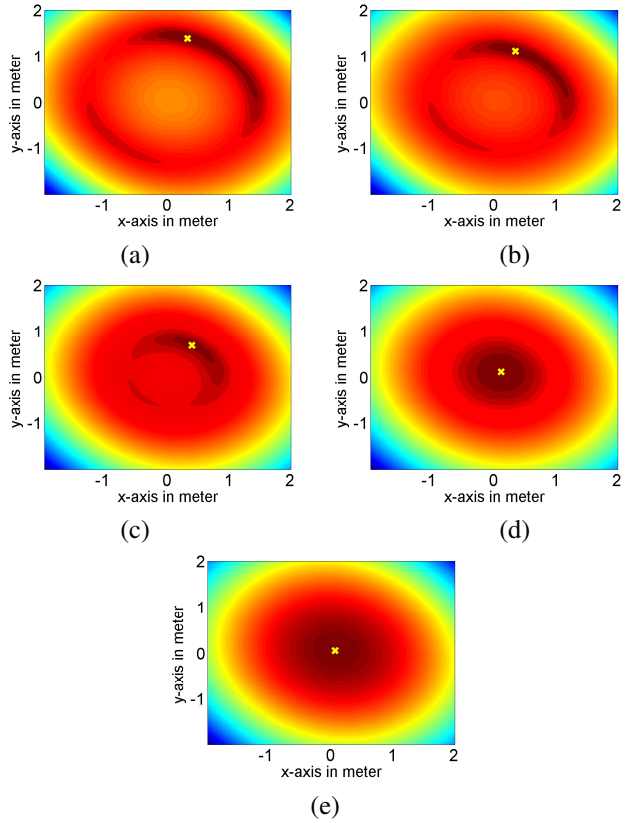

Fig. 11. The cost function as a color level: (a) result for $t_z = 1.6m$, (b) result for $t_z = 1.8m$, (c) result for $t_z = 2m$, (d) result for $t_z = 2.2m$ (e) result for $t_z = 2.4m$

that the proposed method achieves better tracking of the pose for image sequences and also a higher rate of convergence compared to the other methods considered. Future works will be devoted to extend this method to model-free pose estimation and also to camera calibration from a set of points.

REFERENCES

[1] A. Ansar and K. Daniilidis. Linear Pose Estimation from Points or Lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[2] H. Araujo, R. L. Carceroni, and C. M. Brown. A Fully Projective Formulation to improve the Accuracy of Lowe's Pose-Estimation Algorithm. *Computer Vision and Image Understanding*, 70:227–238, 1998.

[3] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'07*, pages 1683–1688, San Diego, CA, USA, 2007.

[4] J. Courbon, Y. Mezouar, and P. Martinet. Evaluation of the unified model on the sphere for fisheye cameras in robotic applications. *Advanced Robotics*, 6(8):947–967, 2012.

[5] D. Dementhon and L. Davis. Model-based Object Pose in 25 Lines of Code. *Int. Journal of Computer Vision*, 15(1-2):123–141, June 1995.

[6] M. Dhome, M. Richetin, J.-T. Lapreste, and G. Rives. Determination of the Attitude of 3D Objects from a Single Perspective View. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, Dec 1989.

[7] P. Fiore. Efficient Linear Solution of Exterior Orientation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):140–148, Feb 2001.

[8] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical applications. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, ECCV '00, pages 445–461, London, UK, UK, 2000. Springer-Verlag.

[9] C. Geyer and K. Daniilidis. Mirrors in Motion: Epipolar Geometry and Motion Estimation. *Int. Journal on Computer Vision*, 45(3):766–773, 2003.

[10] D. Grest, T. Petersen, and V. Krger. *A Comparison of Iterative 2D-3D Pose Estimation Methods for Real-Time Applications*, volume 5575 of *Lecture Notes in Computer Sciences: Image Analysis*, pages 706–715. Springer, 2009.

[11] T. Hamel and R. Mahony. Visual Servoing of an Under-Actuated Dynamic Rigid Body System: an Image-Based Approach. *IEEE Trans. on Robotics and Automation*, 18(2):187–198, April 2002.

[12] J. R. Hurley and R. B. Cattell. The procrustes program: Producing direct rotation to test a hypothesized factor structure. *Systems Research and Behavioral Science*, 7:258–262, 1962.

[13] F. Janabi-Sharifi and M. Marey. A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing. *IEEE Trans. on Robotics*, 26(5):939–947, October 2010.

[14] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP : An Accurate O(n) Solution to the PnP Problem. *Int. Jour. on Computer Vision*, 12(2):155–166, December 2009.

[15] D. G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:441–450, 1991.

[16] C.-P. Lu, G. Hager, and E. Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, Jun 2000.

[17] E. Malis, Y. Mezouar, and P. Rives. Robustness of image-based visual servoing with respect to uncertainties on the 3d structure. *IEEE Trans. on Robotics, I-TRO*, 26(1):112–120, Feb. 2010.

[18] R. Safaee-Rad, I. Tchoukanov, K. Smith, and B. Benhabib. Three-Dimensional Location Estimation of Circular Features for Machine Vision. *IEEE Transactions on Robotics and Automation*, 8(5):624–640, Oct 1992.

[19] P. H. Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10, March 1966.

[20] G. Schweighofer and A. Pinz. Robust Pose Estimation from a Planar Target. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28:2024–2030, 2006.

[21] A. Shademan and F. Janabi-Sharifi. Sensitivity Analysis of EKF and Iterated EKF Pose Estimation for Position-Based Visual Servoing. In *IEEE Int. Conf. on Control Applications*, pages 755–760, Toronto, Canada, August 2005.

[22] O. Tahri and F. Chaumette. Complex Objects Pose Estimation Based on Image Moment Invariants. In *IEEE Int. Conf. on Robotics and Automation, ICRA'05*, pages 438–443, Barcelona, Spain, April 2005.

[23] W. Wilson, C. Hulls, and G. Bell. Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696, October 1996.